What is Claimed is:

1.   A method of optimizing data streaming in a peer-to-peer architecture including a plurality of clients in a chain, the method comprising:

each client monitoring its own bandwidth;

each client informing a succeeding client in the chain of that bandwidth;

each client comparing its own bandwidth with the bandwidth of a preceding client in the chain; and

each client, in response to a difference between the compared bandwidths, reordering its position among the clients in the chain.

2.   The method of Claim 1, wherein each client identifies a preceding client in the chain to the succeeding client in the chain.

3.   The method of Claim 1, wherein a detecting client detects that its bandwidth is greater than that of the preceding client in the chain and, in response, opens a connection with a client upstream of the preceding client.

4.   The method of Claim 3, wherein the detecting client identifies a succeeding client in the chain to the preceding client in the chain.

5.   The method of Claim 4, wherein the preceding client opens a connection with the identified succeeding client.

6.   The method of Claim 3, wherein the or each of the connections is opened concurrently with pre-existing connections between clients in the chain.

7.   The method of Claim 3, wherein after the or each concurrent connection has been made to a client, the or each associated pre-existing connection to that client is dropped.

8.   The method of Claim 7, wherein the client switches to reading local buffer memory before the pre-existing connection is dropped.

9.   The method of Claim 5, wherein, in the reordered chain, the detecting client receives streamed data via the connection from the client that was upstream of the preceding client.

10.   The method of Claim 9, wherein the detecting client sends streamed data to the preceding client.

11.   The method of Claim 10, wherein the pre-existing connection between the preceding client and the detecting client is reversed.

12.   The method of Claim 11, wherein a replacement connection is opened between the preceding client and the detecting client.

13.   The method of Claim 9, wherein, in the reordered chain, the succeeding client receives streamed data via the connection from the preceding client.

14.   The method of Claim 1, wherein after the chain has been reordered, a client synchronizes a timecode of data in local buffer memory with a timecode of data received from a new streamed data input source before switching to data received from that source.

15.   The method of Claim 1, wherein a client replenishes its local buffer memory after the chain has been reordered.

16.   A peer-to-peer data streaming system comprising:
          a plurality of clients in a chain, each client including bandwidth-monitoring means for monitoring its own bandwidth, communication means for informing a succeeding

client in the chain of that bandwidth, comparison means for comparing its own bandwidth with the bandwidth of a preceding client in the chain, and reconfiguration means responsive to a difference between the compared bandwidths to reorder its position among the clients in the chain.

17. The system of Claim 16, wherein a client includes address-providing means for receiving and storing the address of a preceding or succeeding client in the chain and providing that address to, respectively, the succeeding or preceding client in the chain.

18. The system of Claim 16, wherein the comparison means of a client is associated with connection means for receiving the address of, and opening a connection with, a client upstream of the preceding client if the comparison means detects that the bandwidth of its associated client is greater than that of the preceding client in the chain.

19. The system of Claim 18, wherein the connection means is capable of opening a connection concurrently with a pre-existing connection between clients in the chain.

20. The system of Claim 19, wherein the connection means is responsive to making the concurrent connection to drop the associated pre-existing connection.

21. The system of Claim 20, wherein the connection means is associated with switch means for switching the client to read local buffer memory before the pre-existing connection is dropped.

22. The system of Claim 18, wherein the connection means is capable of reversing a pre-existing connection between clients in the chain.

23.   The system of Claim 16, wherein a client comprises data synchronizing means for synchronizing a timecode of data in local buffer memory with a timecode of data received from a new streamed data input source.

24.   The system of Claim 23, wherein a client comprises switch means responsive to the data synchronizing means to switch to data received from the new streamed data input source when the timecodes are synchronized.

25.   A client terminal for use in a peer-to-peer data streaming system having a plurality of client terminals in a chain, the client terminal being configured or programmed to include bandwidth-monitoring means for monitoring its own bandwidth, communication means for informing a succeeding client terminal in the chain of that bandwidth, comparison means for comparing its own bandwidth with the bandwidth of a preceding client terminal in the chain, and reconfiguration means responsive to a difference between compared bandwidths to reorder its position among the client terminals in the chain.

26.   A program storage medium readable by a computer having a memory, the medium tangibly embodying one or more programs of instructions executable by the computer to perform method steps for configuring or programming a client terminal for use in a peer-to-peer data streaming system having a plurality of client terminals in a chain, the method steps comprising the steps of:

configuring or programming the client terminal to monitor its own bandwidth;

configuring or programming the client terminal to inform a succeeding client terminal in the chain of that bandwidth;

configuring or programming the client terminal to compare its own bandwidth with the bandwidth of a preceding client terminal in the chain; and

configuring or programming the client terminal to reorder its position among the client terminals in the chain based upon a difference between compared bandwidths.